

CLAIMS

1. (Currently Amended) In a voice-extensible-markup-language-enabled voice-application deployment architecture, an application logic for determining which portions of a voice application for deployment should be cached at an application-receiving end system or systems, comprising:

a processor for processing the voice application according to sequential dialog files of the application;

a static content optimizer connected to the processor for identifying files containing static content, wherein the static content optimizer tags the files containing static content with a static tag; and

a dynamic content optimizer connected to the processor for identifying files containing dynamic content, wherein the dynamic content optimizer analyzes previous execution of the voice application to determine whether to cache each file containing dynamic content, and wherein the dynamic content optimizer tags the files containing dynamic content with an appropriate dynamic tag;

~~characterized in that the optimizers determine which files should be cached at which end-system facilities, tag the files accordingly, and prepare~~ the files containing static content and the files containing dynamic content ~~those files~~ for distribution to selected end-system cache facilities, based on the tags associated with each content, for local retrieval during consumer interaction with the voice deployed application.

2. (Original) The application logic of claim 1 wherein the static and dynamic optimizers are software routines.

3. (Original) The application logic of claim 1 wherein the static and dynamic optimizers are firmware components embedded into the processor.

4. (Original) The application logic of claim 1 wherein the processor is a dialog runtime processor dedicated to processing subsequent dialogs of a voice application.

5. (Original) The application logic of claim 1 wherein the deployment architecture includes an application server and a voice portal.

6. (Original) The application logic of claim 1 wherein the dynamic optimizer identifies dynamic content according to a determination of non-recurring menu dialog and non-recurring result dialog fetched as a result of consumer interaction with the voice application.

7. (Original) The application logic of claim 1 wherein the cache facility at the end system is a telephony server cache.

8. (Original) The application logic of claim 1 wherein the cache facility at the end system is a Web controller cache.

9. (Original) The application logic of claim 1 wherein the file tagging is accomplished using HTTP 1.1 resource tagging.

10. (Original) The application logic of claim 1 wherein dynamic tagging by the dynamic optimizer uses results from statistical analysis to determine which files to tag for distribution to an end-system cache.

11. (Original) The application logic of claim 1 wherein dynamic optimization continues after application deployment, the continued dynamic tagging relying on changing statistical probability results.